

## AddOrdersWorkflow

Calls OrderManagerFactory to create an ordermanager object

Create Order object to store user info Create variable to store OrderDate

Get order data from user: OrderDate convert to DateTime, Must be in the future

Get CustomerName Can't be blank, a-z,0-9,comma, period

Get State
Call tax list
Has to be in tax file (LINQ contains on response.taxlist)

Get ProductType Show a list of products from file Has to be in product file

Get Area:

Must be positive decimal Minimum order size is 100 sq ft.

Calculate and assign other fields
OrderNumber (Load existing file if there is one & check for max(orderNumber), add 1
TaxRate - from taxes file (linq?)
CostPerSquareFoot = from products file(linq?)
LaborCostPerSquareFoot = from products file( linq?)
MaterialCost = Area \* costPerSquareFoot
LaborCost = area \* laorcostpersquarefoot
Tax ((materialcost + laborcost) \* (taxrate/100))
Total = MaterialCost + LaborCost + Tax

Show Summary of order(call consoleio.display order)

Prompt to confirm

if yes, write data to order file (create new if necessary) with correct date, calling orderManager.AddOrder if no, just return to main menu

## EditOrderWorkflow

Calls OrderManagerFactory to create an ordermanager object

prompt user for date and order number Use lookup to retrieve order to be edited

Create variables store user info - CustomerName, State, ProductType, Area

Get CustomerName
Can't be blank, a-z,0-9,comma, period
if just press enter, skip
if enter new name, save to retrieved order

Get State
if just press enter, skip
if enter new state, call tax list and verify it's in there
Save to retrieved order

Get ProductType
Show a list of products from file
If just press enter, skiip
Has to be in product file
save to retrieved order

Get Area:

if just press enter, skip Must be positive decimal, Minimum order size is 100 sq ft. save to retrieved order

Show Summary of order(call consoleio.display order)

Prompt to confirm if yes, replace old order with new in order file with correct date, calling orderManager.RemoveOrder, then orderManager.AddOrder if no, just return to main menu

Data Entry Validation

Date
CompanyName
State
ProductType
Area
OrderNumber
Calculate Values

## RemoveOrderWorkflow

Calls OrderManagerFactory to create an ordermanager object

prompt user for date and order number Use lookup to retrieve order to be edited & display

Prompt to confirm if yes, remove order from file, calling orderManager.RemoveOrder if no, just return to main menu

Prod & TestTaxRespository: ITaxRepositoy

Loads test or prod tax respository into a list (TaxList) returns LoadTaxFileResponse with list, success, message if fail

Prod & TestProductRespository: IProductRepository

Loads test or prod product respository into a list (ProductList) returns LoadProductFileResponse with list, success, message if fail

TaxRespositoryFactory
Chooses TaxRepository from app.config
returns a repository

ProductRespositoryFactory
Chooses ProductRepository from app.config
returns a repository